

Chapter 11

Online and Offline Agent Self-Improvement

In the pursuit of self-improvement, intelligent agents leverage optimization as both a mechanism for refining individual components—such as prompt design, workflow orchestration, tool utilization, reward function adaptation, and even the optimization algorithms themselves—and as a strategic framework that ensures these individual improvements are aligned toward coherent performance enhancement. For instance, optimizing the reward function and prompt design in isolation might yield conflicting outcomes, but a strategic approach coordinates these optimizations to maintain coherence and maximize overall effectiveness. We categorize self-evolution into two primary paradigms: *online* and *offline* self-improvement. Additionally, we explore *hybrid* optimization strategies that integrate both approaches to maximize efficiency and adaptability.

11.1 Online Agent Self-Improvement

Online self-improvement refers to real-time optimization in which an agent dynamically adjusts its behavior based on immediate feedback. This paradigm ensures that agents remain responsive to evolving environments by continuously optimizing key performance metrics—such as task success, latency, cost, and stability—in an iterative feedback loop. Online self-improvement is particularly effective in applications that require dynamic adaptability, such as real-time decision-making, personalized user interactions, and automated reasoning systems. Key optimization strategies in online self-improvement can be classified into the following four categories: Iterative Feedback and Self-Reflection, Active Exploration in Multi-Agent Systems, Real-Time Reward Shaping, and Dynamic Parameter Tuning.

Iterative Feedback and Self-Reflection These methodologies [48, 67, 72, 70, 847, 47] focus on enabling agents to critique and refine their own outputs iteratively. Reflexion [48], Self-Refine [67], and Tree of Thoughts [72] introduce self-critique loops, where the model identifies errors and proposes revisions in real-time. ReAct [70] combines chain-of-thought “reasoning” with “acting”, allowing the model to revise steps iteratively after observing external feedback. In addition, other methods either rely on self-consistency [78] to select the most coherent solution or leverage a process reward model (PRM) Lightman et al. [847] to choose the best solution from the candidates. Collectively, these frameworks reduce error propagation and support rapid adaptation without requiring a separate offline fine-tuning cycle.

Active Exploration in Multi-Agent Systems These approaches [626, 848, 627, 152] actively explore and dynamically search for novel patterns and workflow improvements in multi-agent systems. MetaGPT [626], CAMEL [848], and ChatDev [627] showcase multi-role or multi-agent ecosystems that interact in real-time, exchanging continuous feedback to refine each other’s contributions. Similarly, HuggingGPT [152] coordinates specialized models (hosted on Hugging Face) through a central LLM controller, which dynamically routes tasks and gathers feedback. These collaborative strategies further highlight how online updates among agents can incrementally refine collective outcomes.

Real-Time Reward Shaping Rather than relying on fixed or purely offline reward specifications, some frameworks [731, 91, 105, 849] integrate immediate feedback signals not only to correct errors, but also to adapt internal reward functions and policies. This enables self-adaptive reward calibration that balances trade-offs between performance, computational cost, and latency, allowing agents to optimize reward mechanisms dynamically in response to user interactions.

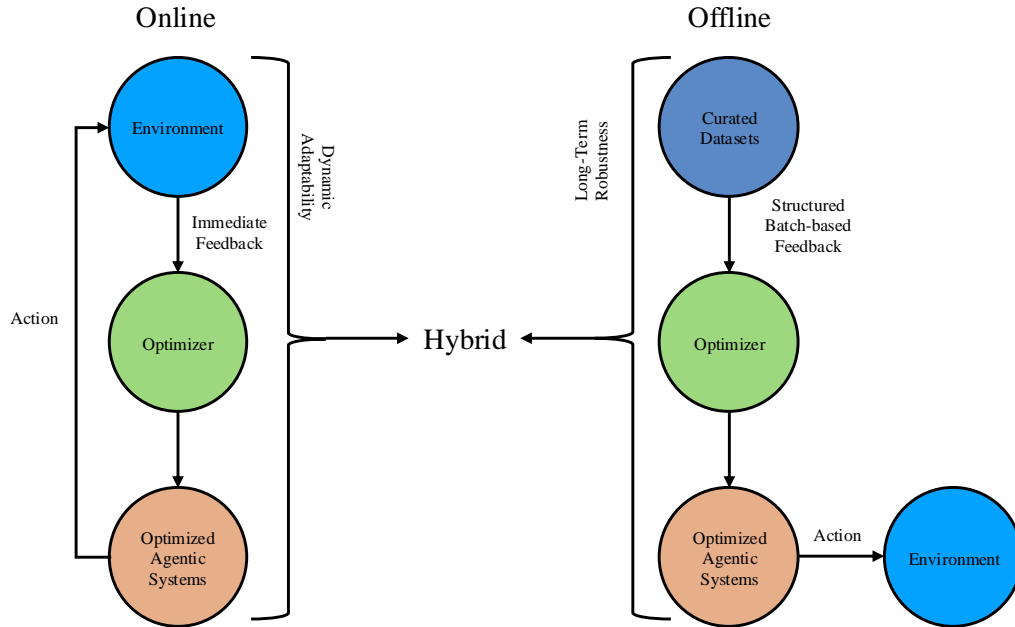


Figure 11.1: An illustration of self-improvement under three different utilization scenarios, including Online, Offline, and Hybrid self-improvement.

Dynamic Parameter Tuning In this category, agents autonomously update their internal parameters (including prompt templates, tool invocation thresholds, search heuristics, etc.) in real time, leveraging gradient-free or approximated gradient methods. These updates optimize both computational efficiency and decision accuracy, allowing for seamless adaptation to evolving contexts. Self-Steering Optimization (SSO) [850] eliminates the need for manual annotation and maintains signal accuracy while keeping training on-policy by autonomously generating preference signals during iterative training.

Online self-improvement fosters a continuously evolving agent framework where learning is embedded within task execution, promoting enhanced real-time adaptability, user-centric optimization, and robust problem-solving capabilities.

11.2 Offline Agent Self-Improvement

Offline self-improvement, in contrast, leverages structured, batch-based optimization. This paradigm utilizes scheduled training sessions with high-quality curated datasets to systematically improve the agent’s generalization capabilities [851, 667, 852, 853, 854]. Unlike online approaches, offline approaches accommodate more computationally intensive methodologies, including Batch Parameter Updates and Fine-Tuning, Meta-Optimization, and Systematic Reward Model Calibration.

Batch Parameter Updates and Fine-Tuning In this category, agents undergo extensive fine-tuning using supervised learning or reinforcement learning (RL) techniques, optimizing performance across large-scale datasets over multiple training epochs. Retrieval-augmented generation (RAG) is often integrated to enhance contextual understanding and long-term memory retrieval [740, 741]. Such methods allow agents to optimize retrieval strategies, thereby improving reasoning over extensive knowledge corpora.

Meta-Optimization of Agent Components Here offline training is not limited to improving task performance but extends to refining optimization algorithms themselves. Meta-learning strategies that optimize hyperparameters or even restructure the optimization process dynamically have demonstrated promising outcomes [731, 91]. These meta-optimization approaches enable agents to discover the most effective learning parameters for new problem domains.

Systematic Reward Model Calibration Offline settings facilitate the precise calibration of reward models, incorporating hierarchical or listwise reward integration frameworks (e.g., LIRE [855]) to align agent behavior with long-term objectives through gradient-based reward optimization. Such calibration ensures that reward functions reflect real-world task complexity, thereby mitigating bias and enhancing generalization.

The structured nature of offline optimization results in a robust agent baseline, whose performance is fine-tuned to optimize stability, efficiency, and computational cost before real-world deployment. Offline training allows for high-fidelity model refinement and is essential for mission-critical applications requiring predictable performance guarantees.

11.3 Comparison of Online and Offline Improvement

Online and offline optimization offer complementary benefits, each excelling in different aspects of self-improvement. Online optimization thrives in dynamic environments, where real-time feedback enables continuous adaptation. It is well-suited for applications that require immediate responsiveness, such as interactive agents, real-time decision-making, and reinforcement learning systems. However, frequent updates may introduce instability or drift, requiring mechanisms to mitigate performance degradation over time.

In contrast, offline optimization emphasizes structured, high-fidelity training using pre-collected datasets, ensuring robust and stable performance before deployment. By leveraging computationally intensive learning methods such as batch training, fine-tuning, and meta-optimization, offline approaches provide strong generalization and long-term consistency. However, they lack the agility of online learning and may struggle to adapt efficiently to novel scenarios without additional retraining. Table 11.1 summarizes the key distinctions between these two paradigms.

Feature	Online Optimization	Offline Optimization
Learning Process	Continuous updates based on real-time feedback	Batch updates during scheduled training phases
Adaptability	High, capable of adjusting dynamically	Lower, adapts only after retraining
Computational Efficiency	More efficient for incremental updates	More resource-intensive due to batch training
Data Dependency	Requires real-time data streams	Relies on curated, high-quality datasets
Risk of Overfitting	Lower due to continuous learning	Higher if training data is not diverse
Stability	Potentially less stable due to frequent updates	More stable with controlled training settings

Table 11.1: Comparison of Online vs. Offline Optimization Strategies in Self-Improvement Agents.

While both approaches have inherent strengths and trade-offs, modern intelligent systems increasingly integrate them through hybrid optimization strategies. These hybrid frameworks leverage the stability of offline training while incorporating real-time adaptability, enabling agents to maintain long-term robustness while continuously refining their performance in dynamic environments.

11.4 Hybrid Approaches

Recognizing that both online and offline methods have inherent limitations, many contemporary systems adopt *hybrid* optimization strategies. These hybrid methods integrate structured offline optimization with responsive online updates to achieve continuous incremental agent enhancement.

Hybrid optimization explicitly supports self-improvement by empowering agents to autonomously evaluate, adapt, and enhance their behaviors through distinct yet interconnected stages:

- **Offline Pre-Training:** In this foundational stage, agents acquire robust baseline capabilities through extensive offline training on curated datasets. This stage establishes essential skills, such as reasoning and decision-making, required for initial autonomous performance. For instance, frameworks such as the one introduced by Schrittwieser et al. [856] illustrate how offline pretraining systematically enhances initial agent capabilities, ensuring subsequent online improvements are built upon a stable foundation.
- **Online Fine-Tuning for Dynamic Adaptation:** Agents actively refine their capabilities by autonomously evaluating their performance, identifying shortcomings, and dynamically adjusting strategies based on real-time feedback. This adaptive fine-tuning stage directly aligns with the agent self-improvement paradigm by allowing real-time

optimization of agent-specific workflows and behaviors, exemplified by Decision Mamba-Hybrid (DM-H) [857], where agents efficiently adapt to complex, evolving scenarios.

- **Periodic Offline Consolidation for Long-Term Improvement:** periodic offline consolidation phases, agents systematically integrate and solidify improvements identified during online interactions. This ensures that incremental, online-acquired skills and improvements are systematically integrated into the agent’s core models, maintaining long-term stability and effectiveness. The Uni-O4 framework [858] exemplifies how this process enables seamless transitions between offline knowledge consolidation and online adaptive improvements.

Hybrid optimization thus explicitly supports autonomous, continuous evolution by seamlessly interweaving structured offline learning with proactive, real-time online adaptation. This cyclical approach equips agents with both immediate responsiveness and stable long-term improvement, making it ideally suited for complex, real-world scenarios such as autonomous robotics, personalized intelligent assistants, and interactive systems.